# Comparison of the quality of Large Language Models with classical approaches for sentiment analysis in Airbnb revews

Ivan Spirin, Peter Gusev

May 2024

**Abstract**

Large Language Models are gaining increasing attention and present new opportunities in the field of natural language processing. Companies like Airbnb can utilize these models to extract valuable insights. This project aims to develop and compare two LLMs, BERT and GPT-3.5, with traditional models, Logistic Regression and Naive Bayes, for binary sentiment analysis of scraped Airbnb review comments. It was hypothesized that LLMs would outperform traditional models due to their advanced linguistic and contextual understanding. The results indicate that GPT-3.5 achieved the highest macro F1 score of approximately 0.951. Interestingly, both Logistic Regression and Naive Bayes performed similarly to or better than the BERT model. `https://github.com/VanoPekkar/airbnb_sentiment_analysis/`.

## 1 Introduction

Airbnb is an online platform offering 7.7 million short-term rentals globally, connecting hosts with guests and aiming to provide authentic community experiences. The reliability of these private rentals necessitates robust quality checks. Electronic Word-of-Mouth, such as online reviews, has become a crucial information source, valued for its credibility over traditional advertising. However, the sheer volume of reviews poses a challenge for analysis. NLP techniques, like BERT and GPT, have advanced the accuracy of deep learning models for such tasks. This paper addresses the challenge of analyzing the growing number of Airbnb reviews by developing and comparing different NLP models as the evaluation of a review being positive or negative seems to be most important for the customer and their purchasing decision. It focuses on sentiment analysis, which identifies opinions and emotions in reviews, hypothesizing that LLMs will outperform traditional models due to their superior contextual understanding and ability to capture nuanced sentiments.

## 1.1 Team

**Ivan Spirin** training and evaluating models, preparing of report
**Peter Gusev** data collecting and preprocessing, preparing of report

# 2 Related Work

Numerous studies have been conducted on Airbnb reviews, primarily because the availability of reviews on their website via scraper tools opens up various research avenues such as sentiment analysis and topic modeling.
Regarding sentiment analysis, [Pouya Rezazadeh Kalehbasti, 2021] used sentiment analysis with TextBlob to extract features from Airbnb reviews, which were then used to predict pricing. Likewise, [Abdelaziz Lawani, 2019] investigated factors influencing the pricing of Airbnb accommodations and found a positive correlation between the positivity of reviews and the price. They employed a lexicon-based model to calculate the sentiment of a review by summing the positivity or negativity of each word. [M. R. Raza and Varol, 2022] compared basic RNN, LSTM and GRU for sentiment classification of Airbnb reviews. Their results indicated that GRU had the highest accuracy and other performance metrics.
In the context of advanced deep learning models, [Qihuang Zhong, 2023] compared ChatGPT which is based on GPT-3.5 with four fine-tuned BERT-style models. They found that while ChatGPT did not perform as well in handling paraphrases and similarity tasks, it outperformed the fine-tuned BERT models in inference tasks. For sentiment analysis, both models achieved similar performance levels.

# 3 Model Description

All four models described below predict the negative or positive sentiment based on train data with pre-existing labels.

## 3.1 Naive Bayes

This approach utilizes Bayes' theorem to classify text. During the training phase, NB calculates the probability of each unique word for each class based on its observed frequencies in the training data to make predictions. Laplace smoothing, which adds one to the count of each word in the training corpus to handle unseen words, was applied, which resulted in the best performance of this model.

## 3.2 Logistic Regression

Logistic Regression assigns a weight to each feature and adds a bias term. The resulting score is mapped utilizing the sigmoid function to assign a class to each review.

## 3.3 BERT

For the deep learning model, the DistilBERT adaptation by Sanh et al. (2020) was chosen due to its strong performance on various NLP tasks and its ease of adaptation using transfer learning techniques. A custom classifier was trained on top of the features extracted from the base DistilBERT model. BERT was pretrained on two massive language datasets: the BooksCorpus by [Yukun Zhu, 2015] and English Wikipedia pages, totaling over 3 billion words. Unlike GPT, BERT was trained to handle whole sequences rather than individual words to better understand language structure. BERT's pretraining involves two key techniques. First, the masked language model approach randomly hides 15% of tokens in the input, requiring the model to predict these masked tokens based on the surrounding context. Second, the model is trained for Next Sentence Prediction, determining if one sentence follows another from the input dataset. [Victor Sanh, 2019] distilled the BERT model to create DistilBERT, reducing its size and complexity. This was achieved by training a smaller "student" network to replicate the behavior of a larger "teacher" network, in this case, the original BERT-base model with 110 million parameters. The student model, with half the number of layers and lacking the token-type and pooling layers of BERT, has 66 million parameters while retaining 97% of the original model's performance.

The classifier configuration follows a standard methodology of progressively reducing the number of neurons towards the end of the network. A classifier with the same number of neurons per block was tested but performed slightly worse than one with a 50% reduction in neurons per layer. The dropout rate, activation function, and kernel initialization were determined using a grid search. The tanh activation function, while sacrificing a small amount of precision, provided better recall and F1 score performance compared to the more commonly used ReLU activation, and was thus chosen. Classifier architecture is presented in Table 1.

| Layer | Activation | Parameters | Initialization |
|---|---|---|---|
| Dense, 256 | Tanh | 196864 | he normal |
| Dense, 128 | Tanh | 32892 | he normal |
| Dropout, 10% | - | 0 | - |
| Dense, 64 | Tanh | 8256 | he normal |
| Dense, 32 | Tanh | 2080 | he normal |
| Dropout, 10% | - | 0 | - |
| Dense, 2 | Softmax | 64 | he normal |

Table 1: classifier part architecture

### 3.4 GPT-3.5

This paper utilizes the GPT-3.5 “gpt-3.5-turbo-instruct” model, a specialized variant of GPT with 175 billion parameters. The model’s performance was assessed using a randomly selected sample of 1,000 reviews from the test set. This sample size was chosen primarily due to API requests to OpenAI were quite time consuming. GPT-3.5 is pre-trained, eliminating the need for explicit training on a sentiment-labeled review dataset. Instead, it leverages its pre-training to interpret the sentiment of text based on patterns learned from a vast dataset. To analyze each review and predict its sentiment, the model was prompted with a following query:
*Given the following Airbnb review, predict the reviewer’s rating as 1 or 0, where 1 is positive and 0 is the negative. Provide your answer as only an integer. Here is the review: ‘review’*
The predicted sentiment was extracted from the model’s response and recorded. To handle possible errors or timeouts during API calls, the process included retries. After processing all reviews, a dataframe for evaluating the model’s performance was created: it contains the review comment, the predicted sentiment, and the actual sentiment for each review.

## 4 Dataset

### 4.1 Collecting

Raw data was collected using the Airbnb Scraper application via Apify website. Since this tool collects information directly from the Airbnb website, the data obtained will be current at the time of the study. Due to restrictions on query size, data was collected in several iterations for different cities and price ranges. For the study, several large European cities were selected for which more than 1000 active advertisements were available: London, Paris, Dublin, Berlin, Munich. Raw data consists of 7492 advertisments and 311785 reviews.

### 4.2 Preprocessing

Primary data cleaning includes removing duplicates that could have been obtained during data collection, since the data was scraped not in a single iteration. After this, reviews written in languages other than English were removed. Their number constituted approximately 10% of the sample, but was large enough to translate into English. In addition, translation from some languages may not be accurate enough, which would degrade the quality of the data. Reviews for advertisements with empty values for some parameters were removed, and a study was also conducted to identify price outliers and reviews for advertisements with unrealistic values were removed. This was done to exclude possibly fake advertisements from the sample.
At the second stage of preprocessing, the text of reviews was transformed: line breaks, special characters, numbers, locations, and names were removed from

the text. Also, all words were converted into lower cases. Target transformations were made. Firstly, reviews with a zero rating have been eliminated, since these are reviews generated by Airbnb itself. Also, the five-rank assessment system was converted to binary. Reviews with a rating of 4 or 5 received a positive label, and reviews with ratings of 1 and 2 received a negative label. Reviews with a rating of 3 were not considered, since they are difficult to clearly classify as positive or negative. The distribution of targets for the final data is presented on Figure 1.

At the third stage for Naive Bayes and Logistic Regression models, lemmatization and stemming were applied to reduce word variations and normalize the text data. By consolidating similar words and reducing them to their base or root forms, the core semantic content was preserved while minimizing redundancy. Additionally, stopwords were removed to eliminate common words that provide little discriminatory information, allowing a focus on more meaningful terms. However, some negators such as ‘not’ were retained. A filtering step was implemented to remove infrequently occurring words that appeared fewer than 10 times. Finally, the reviews were tokenized into unigrams, bigrams, and trigrams. As a result, the dimension of the document-term matrix is reduced, aiming to balance retaining meaningful words with removing those that do not significantly contribute to sentiment analysis. This also makes the resulting vectors for each review less sparse. For BERT model just pretrained tokenizer was applied to primarily preprocessed data and to GPT-3.5 the text data was passed via API calls.

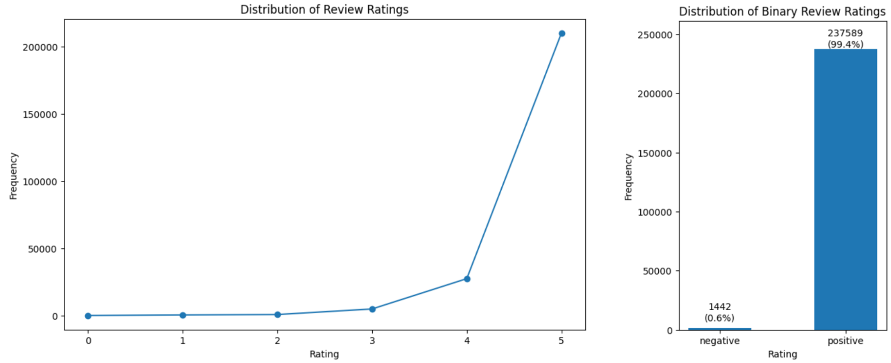

Figure 1: target distribution

# 5 Experiments

## 5.1 Metrics

Since the classification problem was being solved, it was decided to use precision, recall and F1 score as metrics. Due to the large imbalance of classes, special attention was paid to the value of these metrics in the negative class. For this reason accuracy metric is also not indicative.

## 5.2 Experiment Setup

**Data split** As can be seen in Figure 1, the data distribution is strongly skewed towards positive labels. In order to balance the classes, two oversampling techniques, random oversampling and SMOTE, were used for the test and validation sets. For each of the four models, a technique was chosen that allowed us to obtain the best quality on the test sample. The dataset was randomly divided into a training (60%), validation (20%), and test (20%) set, final sizes are presented in Table 2.

| | Train | Valid | Test |
|---|---|---|---|
| Before oversampling | 191,224 | 63,741 | 63,741 |
| After oversampling | 380,140 | 126,207 | 63,741 |

Table 2: datasets' sizes

**Finetuning and hyper-parameters** For the Bayes Classifier optimal smoothing parameter was selected in a grid search with 5-fold cross-validation.
For the Logistic Regression the grid-search with 5-fold cross validation was applied to select the optimal regularization parameter to limit overfitting.
Fintuning was used for the DistilBERT model. To save training time and develop the classifier faster, the weights of the main part were frozen and only the weights in the classifier part were trained. To ensure the best model from the training, each time the validation loss improves, the current model weights are saved. The training configuration is shown in Table 3.

| Parameter | Value |
|---|---|
| Optimizer | Adam |
| EarlyStopping | 64 epoch |
| Learning rate | 0.001 |
| LR reduction | 0.1 factor, 16 ep patience |
| Batch size | 512 |

Table 3: DistilBERT training parameters

## 5.3 Baselines

Due to its simplicity and reliability, the models described in sections 3.1 and 3.2, Naive Bayes and Logistic Regression, could be considered as a baselines.

They were compared with LLMs, which were supposed to show better quality on the proposed task.

# 6 Results

Results for all models are presented in Table 4. All models have a similar accuracy of about 0.99. Overall, GPT outperformed all other models with a macro F1 score of approximately 0.951. The GPT model shows a strong balance between precision (0.852) and recall (0.966), effectively addressing the weaknesses of the other models discussed in this paper. Logistic Regression slightly surpasses Naive Bayes with a macro F1 score of 0.801 compared to 0.798. The more complex BERT model has the lowest macro F1 score of 0.74. There is no significant performance difference for the positive class, but there is for the negative class. Naive Bayes and Logistic Regression have lower precision, 0.5 and 0.52, respectively, but compensate with high recall values, 0.75 and 0.719, respectively, capturing a large portion of the true negative class. In contrast, the BERT model has better precision but lower recall. Despite its lower macro F1 score, BERT should not be considered inferior. Its higher precision indicates a better understanding of the negative class's features, leading to more confident predictions for this class. However, BERT's lower recall means that it misses many negative instances that simpler models capture better.

| Label | Precision | Recall | $\mathrm{F}_1 Score$ |
|---|---|---|---|
| Naive Bayes, macro $\mathrm{F}_1$ : 0.7981 | | | |
| Negative | 0.4988 | 0.7500 | 0.5992 |
| Positive | 0.9985 | 0.9954 | 0.9970 |
| Logistic Regression, macro $\mathrm{F}_1$ : 0.8017 | | | |
| Negative | 0.5241 | 0.7188 | 0.6061 |
| Positive | 0.9983 | 0.9960 | 0.9972 |
| BERT model, macro $\mathrm{F}_1$ : 0.7399 | | | |
| Negative | 0.6285 | 0.3935 | 0.484 |
| Positive | 0.994129 | 0.997740 | 0.995931 |
| GPT model, macro $\mathrm{F}_1$ : 0.95157 | | | |
| Negative | 0.85294 | 0.96667 | 0.90625 |
| Positive | 0.99896 | 0.99481 | 0.99688 |

Table 4: Results

# 7 Conclusion

Contrary to expectations, Logistic Regression and Naive Bayes performed similarly or even better than the fine-tuned BERT model. This may be due to the dataset's size or complexity. BERT excels with large, complex datasets, so it might not have utilized its full capabilities here. On the other hand, NB's assumption of conditional independence between features limits its ability to capture complex language dependencies. The task's simplicity could also play a role. Binary sentiment analysis might not need the advanced representations that BERT offers. Additionally, the imbalanced dataset might have affected BERT's learning. As anticipated, GPT-3.5 significantly outperformed all other models in detecting negative sentiment. This success can be attributed to its 175 billion parameters, which provide much greater complexity compared to other models. The combination of this complexity, OpenAI's server infrastructure, and extensive human-supervised fine-tuning likely enhances its performance. These findings indicate that model selection should consider the task's characteristics and the available data. While BERT and GPT-3.5 represent state-of-the-art performance in many NLP tasks, there are situations where simpler models like Logistic Regression and NB can achieve comparable or superior results. These simpler models also offer advantages in terms of lower resource requirements, shorter runtimes, and reduced costs.

# References

[Abdelaziz Lawani, 2019] Abdelaziz Lawani, Michael R. Reed, T. M. Y. Z. (2019). Reviews and price on online platforms: Evidence from sentiment analysis of airbnb reviews in boston. In *Regional Science and Urban Economics*, pages 22–34.

[M. R. Raza and Varol, 2022] M. R. Raza, W. H. and Varol, A. (2022). Performance analysis of deep approaches on airbnb sentiment reviews. In *10th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–5.

[Pouya Rezazadeh Kalehbasti, 2021] Pouya Rezazadeh Kalehbasti, Liubov Nikolenko, H. R. (2021). Airbnb price prediction using machine learning and sentiment analysis. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 173–184.

[Qihuang Zhong, 2023] Qihuang Zhong, Liang Ding, J. L. B. D. D. T. (2023). Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. In *arXiv:2302.10198*.

[Victor Sanh, 2019] Victor Sanh, Lysandre Debut, J. C. T. W. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *http://arxiv.org/abs/1910.01108*.

[Yukun Zhu, 2015] Yukun Zhu, Ryan Kiros, R. Z. R. S. R. U. A. T. S. F. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *https://arxiv.org/abs/1506.06724*.